

AD-A168 359

FUNCTIONAL AGGREGATION AND MIXTURE FORMULATION(U)

1/1

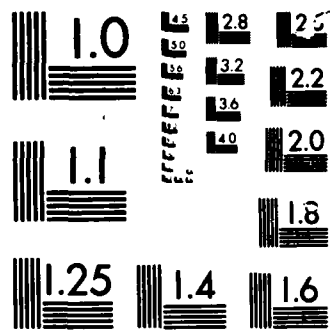
GEORGIA INST OF TECH ATLANTA PRODUCTION AND
DISTRIBUTION RESEARCH CENTER A V IYER ET AL. 1986

UNCLASSIFIED

PDRG-86-84 N00014-85-C-0797

F/G 15/5

NL



MICROCOPY

CHART

Functional Aggregation

and MRIMATE formulation

Ananth. V. Iyer
John J. Jarvis
H. Donald Ratliff

✓
PDRC- 86-04

port by:

Report for:
Joint Deployment Agency
MacDill Air Force Base, FL 33608

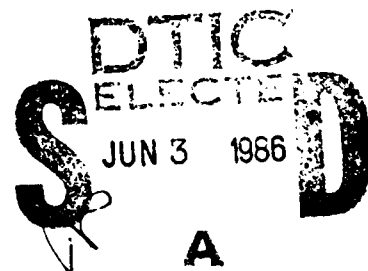
12

Functional Aggregation
and MRMATE formulation

Ananth. V. Iyer
John J. Jarvis
H. Donald Ratliff

PDRC- 86-04

Report by:
School of Industrial and Systems Engineering
Georgia Institute Of Technology
Atlanta, GA 30332



This work is supported by the Office of Naval Research under Contract No. N00014-85-C-0797 and N00014-83-K-0147. Reproduction in whole or part is permitted for any purpose of the U. S. Government.

This document has been approved for public release and sale; its distribution is unlimited.

CONTENTS

1. INTRODUCTION	1
2. MRMATE STRUCTURE AND THE NEED FOR AGGREGATION	2
2.1 MRMATE and its role in SCOPE (MODES)	2
2.2 Network structure of MRMATE	2
2.3 Motivation for aggregation in MRMATE	3
2.3.1 Problem Size	5
2.3.2 Co-ordination	5
2.3.3 Data Quality	5
2.3.4 Modeling Issues	6
3. AGGREGATION DIFFICULTIES IN MRMATE	7
3.1 Aggregation Approaches	7
3.1.1 Fixed weight disaggregation	10
3.1.2 Optimal disaggregation	10
3.2 Sparsity of MRMATE structure	13
3.2.1 Adjusting costs	14
3.2.2 Adjusting Capacities	14
3.2.3 Algorithmic Requirement	18
4. TYPES OF MRMATE AGGREGATION	25
4.1 Aggregation over time	27
4.1.1 Example Problem	28
4.2 Aggregation by mode	32
4.2.1 Example Problem	33
4.3 Aggregating channels within a time period	33
4.3.1 Example Problem	37
5. CONCLUSIONS	42
6. BIBLIOGRAPHY	43

List of Figures

1 Cargo and Channel Types	4
2 Detailed Network Example	8
3 Aggregate Network Example	9
3(a) Cluster 1 Disaggregation	11
3(b) Cluster 2 Disaggregation	12
4 Cost Adjustment Example	15
5 Aggregate Problem with Big-M Costs	16
6 Disaggregation Of Cluster Node 7	17
7(a) Detailed Arcs in a Sparse Network	19
7(b) Aggregate Arc Capacity	20
8 Detailed Network	21
9 Aggregate Network with Arc Capacities	22
10 Disaggregation of Cluster Node 8	23
11 MRMATE Example	26
12 Aggregation Over Time	29
13(a) Channel 1 Disaggregation	30
13(b) Channel 2 Disaggregation	30
13(c) Channel 3 Disaggregation	31
13(d) Channel 4 Disaggregation	31
14 Aggregation by Mode	34
15(a) Disaggregation Problem for Air Channels	35
15(b) Disaggregation Problem for Sea Channels	36
16 Aggregating Channels by Mode within a time period	38
17(a) Disaggregation Problem for Air at $t=1$	39
17(b) Disaggregation Problem for Air at $t=2$	39
17(c) Disaggregation Problem for Sea at $t=1$	40
17(d) Disaggregation Problem for Sea at $t=2$	40

1.0 INTRODUCTION

The SCOPE (and MODES) system consists of the LIFTCAP and MRMATE problems linked via sensitivity information. PDRC report 85-06 discussed a re-formulation of MRMATE as an approach to generate satisfactory solutions to the MRMATE problem at each iteration.

→ This report outlines application of these ideas to the solution of MRMATE. Specifically, the effect of aggregation across time, channels, and MRs is analyzed. This report includes a description of the network structure of MRMATE, the significance of MRMATE nodes, arcs and the limits and costs on them. It also outlines various reasons for the aggregation of MRMATE.

An example MRMATE problem, provides a better understanding of the type of data required in the setup of MRMATE and the types of logical aggregation parameters that could be considered. Effects of the aggregation parameter on problem formulation are discussed and illustrated for the example problem considered.

Accession For	
NTIS	ORNL
DTIC	TRD
Unpublished	
Justification	
By	
Distribution	
Availability	
Dist	Spec

41



2.0 MRMATE STRUCTURE AND THE NEED FOR AGGREGATION

2.1 MRMATE and its role in SCOPE (MODES):

The MODES deployment system consists of the two problems LIFTCAP and MRMATE which interact to create channel configurations and movement requirement transportation plans for a deployment scenario. Descriptions of LIFTCAP and MRMATE and their details are provided in Chapter 4 of PDRC Report 84-09. This section provides a brief description of MRMATE in the development of a deployment scenario.

MRMATE accepts a set of channels configured by LIFTCAP and time expands them according to the planning increments established by the modeler. It then creates a network model from information regarding cargo categories, channels for a cargo type, the time window for a MR at its destination, and the priority ranking of MRs. The MRMATE model then allocates MRs to channels so that as many of the MRs as possible arrive at their destinations within desired time windows.

The solution to MRMATE is fed back to LIFTCAP which, in turn, re-configures channels in order to move towards global optimality of the overall solution to the deployment scenario.

2.2 Network structure of MRMATE:

The MRMATE model is created after LIFTCAP provides a channel configuration. MRMATE accepts movement requirement data regarding cargo type, quantity, point of origin, destination, and time window at the destination. It also receives channel types and their capabilities. Each channel represents a particular mode and route from a POE (point of embarkation) to a POD (point

of disembarkation).

The MRMATE model is a transportation problem with 'source' nodes representing movement requirements; the 'supply' represents the quantity to be transported. 'sink' nodes represent channel capability over time. Each LIFTCAP derived channel is represented by T nodes in MRMATE, where T represents the number of planning periods being modeled. The capacity of each MRMATE channel node is the capability generated by LIFTCAP factored over the number of days the MRMATE channel node represents. The transportation problem 'arcs' represent feasible allocations of movement requirements to channels. A generic model of cargo and asset type allocation restrictions is shown in Figure 1. In this example the restrictions mean that there are arcs only between outsized cargo and outsized channels; oversized cargo and outsized and oversized channels; and bulk cargo and outsized, oversized, and bulk channels.

The objective in this example is to deliver the movement requirement requirements to their required destinations within the time windows specified. This objective is modeled by a (convex) cost function which places a cost on a MR when it is allocated to a channel outside the MR's time window. The cost increases with the distance from the window. A low cost (often zero) is applied to movements scheduled within the window. The objective is to attempt to 'push' MRs towards the center of the time window.

2.3 Motivation for aggregation in MRMATE

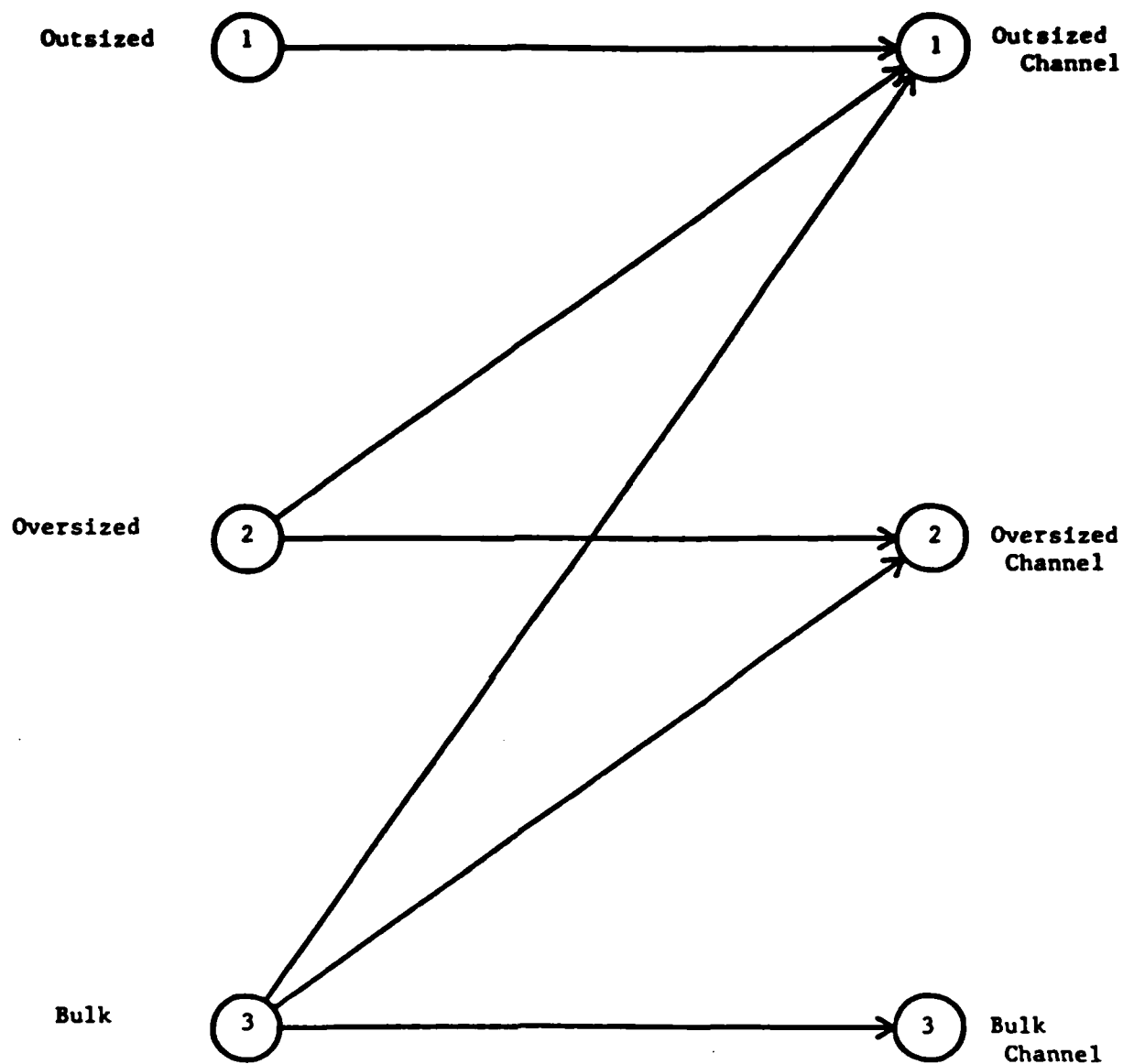


Figure 1. Cargo & Channel Types

2.3.1 Problem Size

Consider the following issue;

Let T = number of time periods,

R = number of movement requirements,

A = number of assets,

I = number of POEs, and

J = number of PODs.

With this notation, MRMATE potentially has R sources, $A * I * J * T$ sink nodes and $R * A * I * J * T$ arcs. For a problem with $A = 10$ assets, $I = 50$ POEs, $J = 50$ PODs, $R = 1000$ MRs and $T = 30$ time periods, MRMATE would have 1000 sources, 0.75 million sinks and potentially 750 million arc variables.

The time required for a solution procedure, based on this model, would preclude its usage in a crisis action deployment situation, particularly since it must be re-solved at each iteration of the SCOPE procedure.

2.3.2 Co-ordination

POEs occur at different zones in the U.S. and are therefore are controlled by different regional transportation controllers. If, at each stage, a deployment plan that is 'acceptable' is desired, then inputs from each of the regions regarding the shipments from that region are necessary. In this case, the MRMATE problem must be solved jointly by the different regional controllers, in co-ordination with the Supporting Commander.

2.3.3 Data Quality

In general, the data available is usually subject to some

degree of uncertainty. Since aggregated data tends to be far more stable than the detailed data values, a deterministic model at an aggregate level is more realistic than a detailed one for uncertain data. Once the aggregate model is solved, disaggregation models could use a human interface to analyze the problem to any required degree of satisfaction.

2.3.4 Modeling Issues

There are usually non-quantifiable constraints and objectives which are not included in the model. An example might be unit integrity. After the problem is solved, the aggregate/disaggregate model enables examination of alternate optimal solutions which might be more satisfactory when these additional constraints are included.

These issues suggest the use of a multi-level procedure which decides on a global MR allocation to zones. This problem could utilize the detailed plan generated by the zonal controllers to modify the global allocation, thereby moving the solution towards a satisfactory objective.

3.0 AGGREGATION DIFFICULTIES IN MRMATE

3.1 Aggregation Approaches

Solution techniques for large scale problems emphasize approaches which work with only a portion of the problem at a time. These methods are not particularly affected by increase in the problem size, except for an overall increase in the time required for solution. Aggregation as an approach for the solution of large scale transportation problems was first examined by Balas [1]. He suggested setting up of an aggregate transportation problem by combining similar source and sink nodes to form aggregate nodes. Procedures for decisions regarding which nodes to combine were left to specifics of the problem under consideration. The aggregate solution was disaggregated to yield a solution to the original problem. The iterative step was based on the dual infeasible arcs in the disaggregated problem. A detailed network example in Figure 2 along with its nodes to be aggregated yields the aggregated problem in Figure 3.

Lee[3] examined the case of a minimum cost flow on a general network and extended Balas's ideas to the general network case. Zipkin[5] considers generation of bounds on the 'loss of information' due to aggregation. He shows how the choice of the components in an aggregate cluster could affect the quality of the bounds generated for a general linear programming problem. He also examines the use of aggregation as a tool to setup equivalent formulations of the original linear programming problem, and to initiate the gradual introduction of detail into the problem by iteratively changing weights used in the

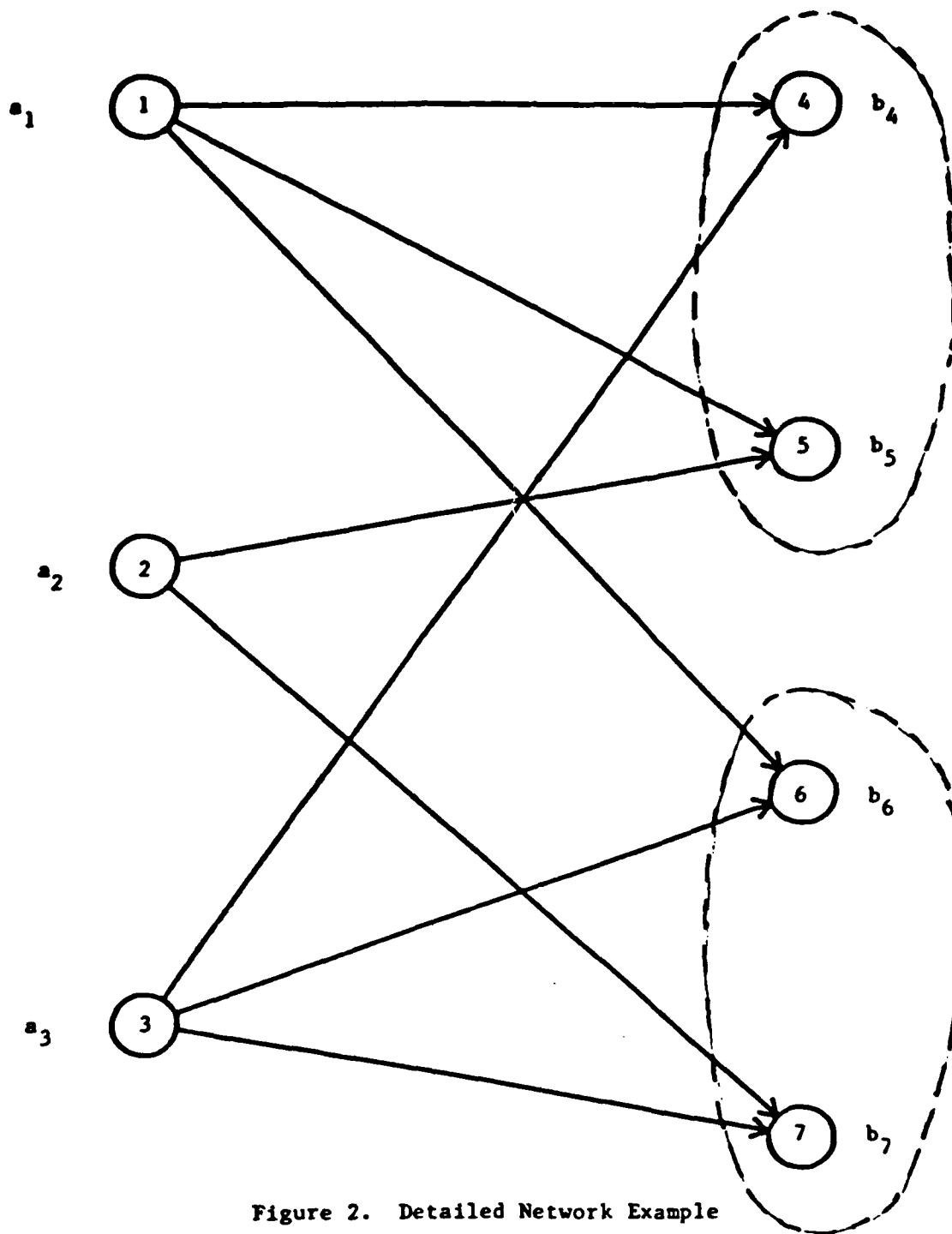


Figure 2. Detailed Network Example

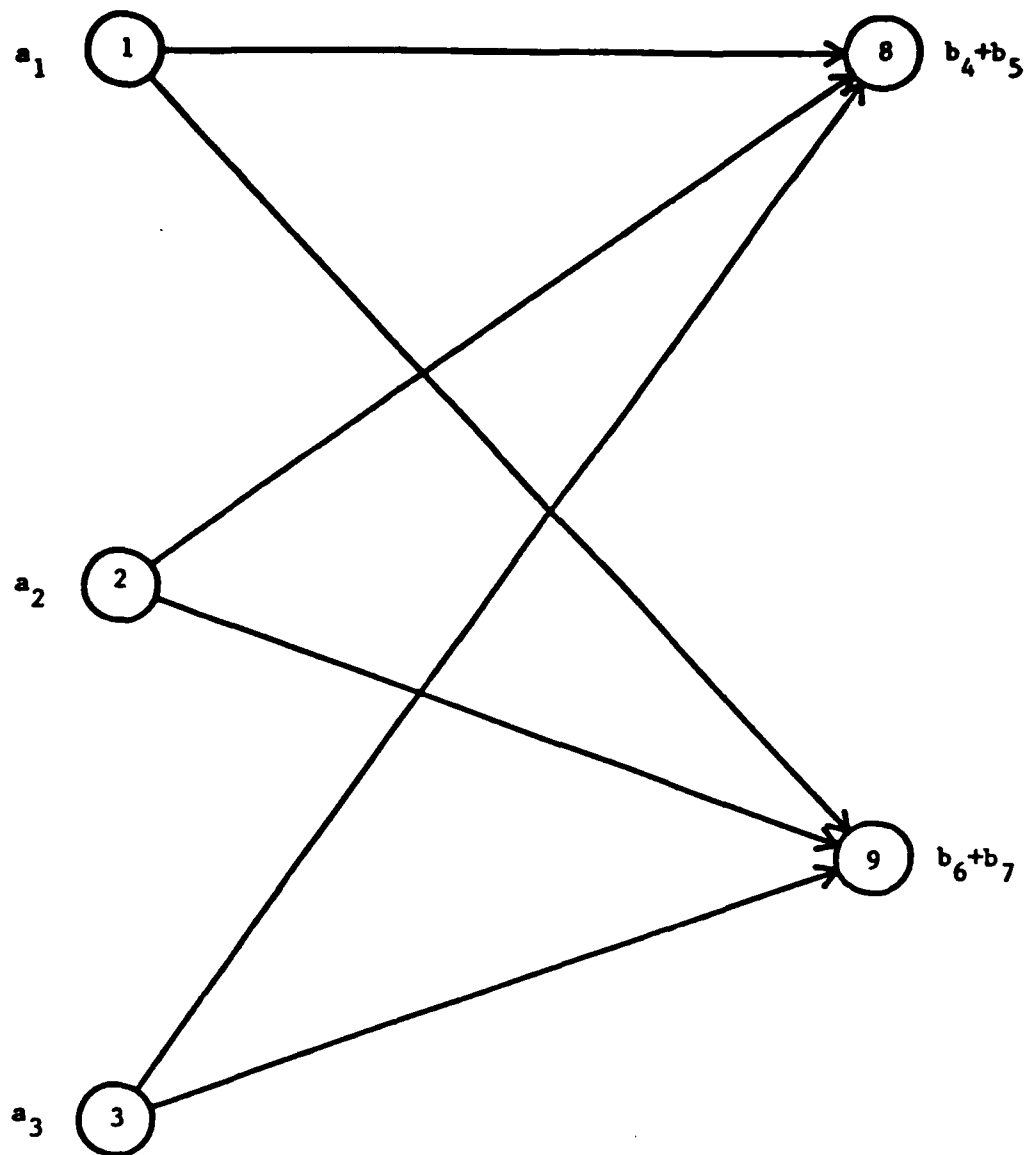


Figure 3. Aggregate Network Example

aggregation. Zipkin also discusses various methods for deriving upper and lower bounds to the solution of the detailed problem at each iteration of the procedure.

Taylor[4] quantified the basic ideas in Zipkin[5] and Geoffrion[2] regarding the choice of components to be aggregated. He establishes a 'closeness' measure between constraints of an linear program. This measure reduces the relation between pairs of constraints to a number between zero and one. It is used to decide which constraints are to be aggregated into clusters, so as to maximize the information available at the aggregate problem level for a given aggregate problem size.

The aggregate problem set up provides a solution which must be disaggregated to provide a solution to the detailed problem. The two basic approaches are as follows

3.1.1 Fixed weight disaggregation

Fixed weight disaggregation essentially multiplies an aggregate solution by a fraction to yield the flows on the detailed arcs. In the case of the complete transportation problem (one with all arcs present between the two node sets) this method can be shown to yield a feasible solution at all times if the multipliers are in proportion to the supply on the incident node as a fraction of the total aggregate node supply.

3.1.2 Optimal disaggregation

The aggregate flows are used to setup independent network flow problems for each cluster. The aggregate flows into each clustered node supply flows to the detailed arcs in the original

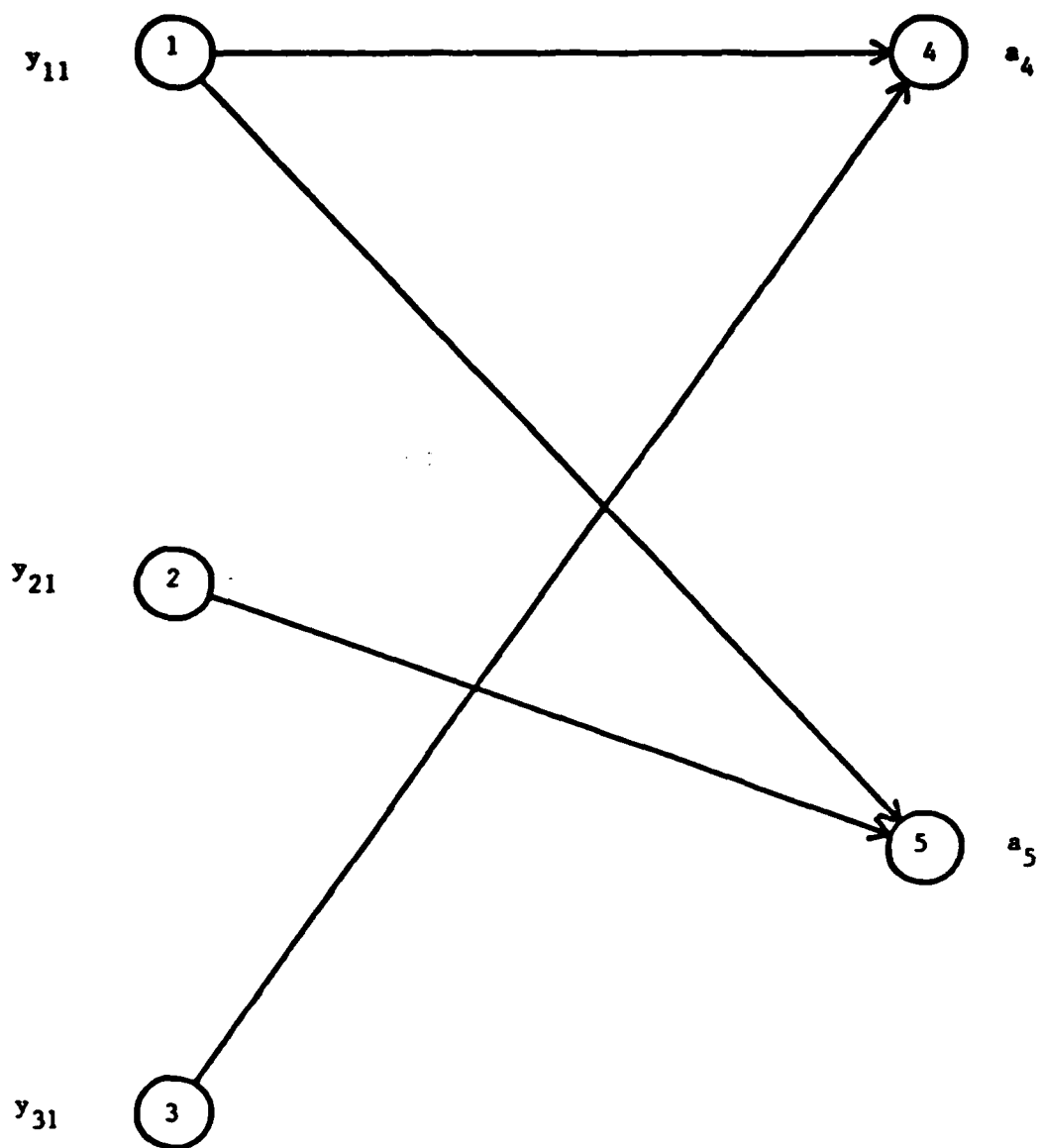


Figure 3(a). Cluster 1 Disaggregation

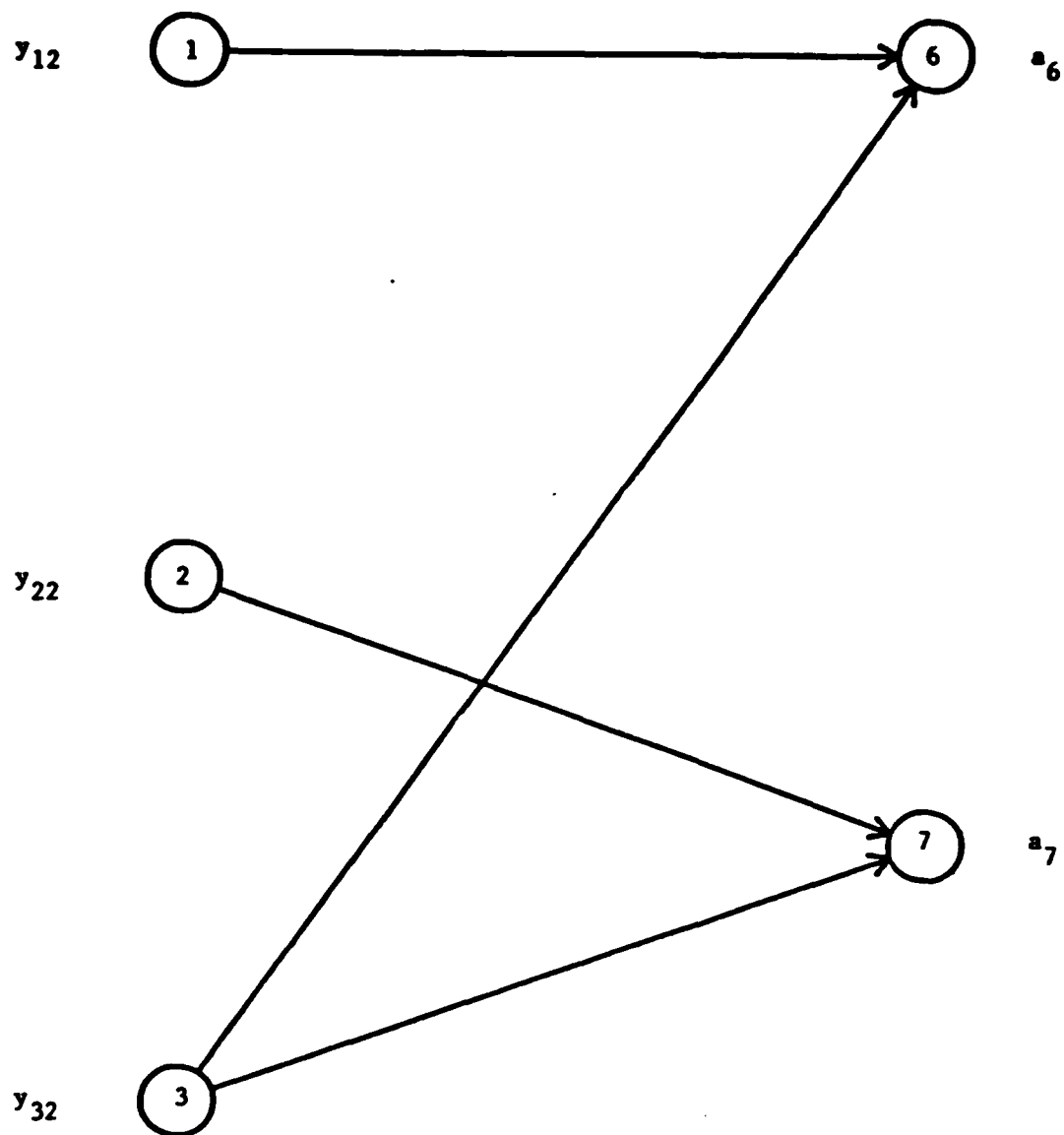


Figure 3(b). Cluster 2 Disaggregation

problem as shown in Figure 2. The effect of the aggregation and optimal disaggregation is to separate the assignment of flows into the detailed arcs in each aggregate node. Arcs in Figure 3(a) are the detailed arcs between the source nodes and the nodes in cluster 1 while arcs in Figure 3(b) are the detailed arcs between source nodes and the nodes in cluster 2.

3.2 Sparsity of the MRMATE structure

When aggregation is used as a solution procedure for the sparse transportation problem, the disaggregated solution cannot be guaranteed to provide a feasible solution to the original problem. A complete transportation problem (with all the arcs present between source and sink nodes) can be assured to be feasible as long as the sum of supplies equal the sum of the demands. This is not, however, true in the general sparse case.

The MRMATE problem with cargo types and different channel types, (i.e. bulk, oversized and outsized categories) is a sparse transportation problem. Assigning outsized cargo to an oversized or bulk channel is an infeasible solution to MRMATE.

The example in Figure 4 shows a sparse transportation problem. A large cost (big-M) is placed on non-existent arcs. Application of Fixed-weight disaggregation in this case is not practical since it ignores the arc costs in disaggregating. It would always try to send flow on the artificial arcs (the ones with cost of big-M). Optimal disaggregation attempts to identify a feasible flow, if it exists. In this case, optimal disaggregation may not identify a feasible solution since the subproblems set up may be infeasible. The infeasibility in this

case results from the fact that the aggregate problem does not have sufficient information about sparsity of the original problem.

Various iterative approaches can be envisaged to handle sparsity.

3.2.1 Adjusting Costs

An approach is to set costs on non-existent arcs. If the costs are too small, the transportation algorithm will not understand that this arc does not really exist. On the other hand, very large costs would tend to inflate the pro-rated cost on the aggregate arc, and may still generate an infeasible solution. An example problem illustrating this case is presented in Fig 4. A large cost (big-M) is placed on the non-existent arcs 1-5 and 2-4. The aggregate problem with pro-rated costs is in Figure 5. However, the disaggregation problem may be infeasible. A solution to the aggregate problem which sends a flow of 0 units on arc 1-7 and 15 units on arc 2-7 would set up an infeasible disaggregation problem as in Figure 6.

An algorithm might proceed as follows. Set costs and solve the aggregate problem. If any of the infeasible arcs have flow in the optimal solution, modify the cost just enough to cause a pivot to occur. The principle issue is how to provide for the reduction of arc costs when the arc flow goes to zero.

3.2.2 Adjusting Capacities

Capacities may be established for arcs in the aggregated

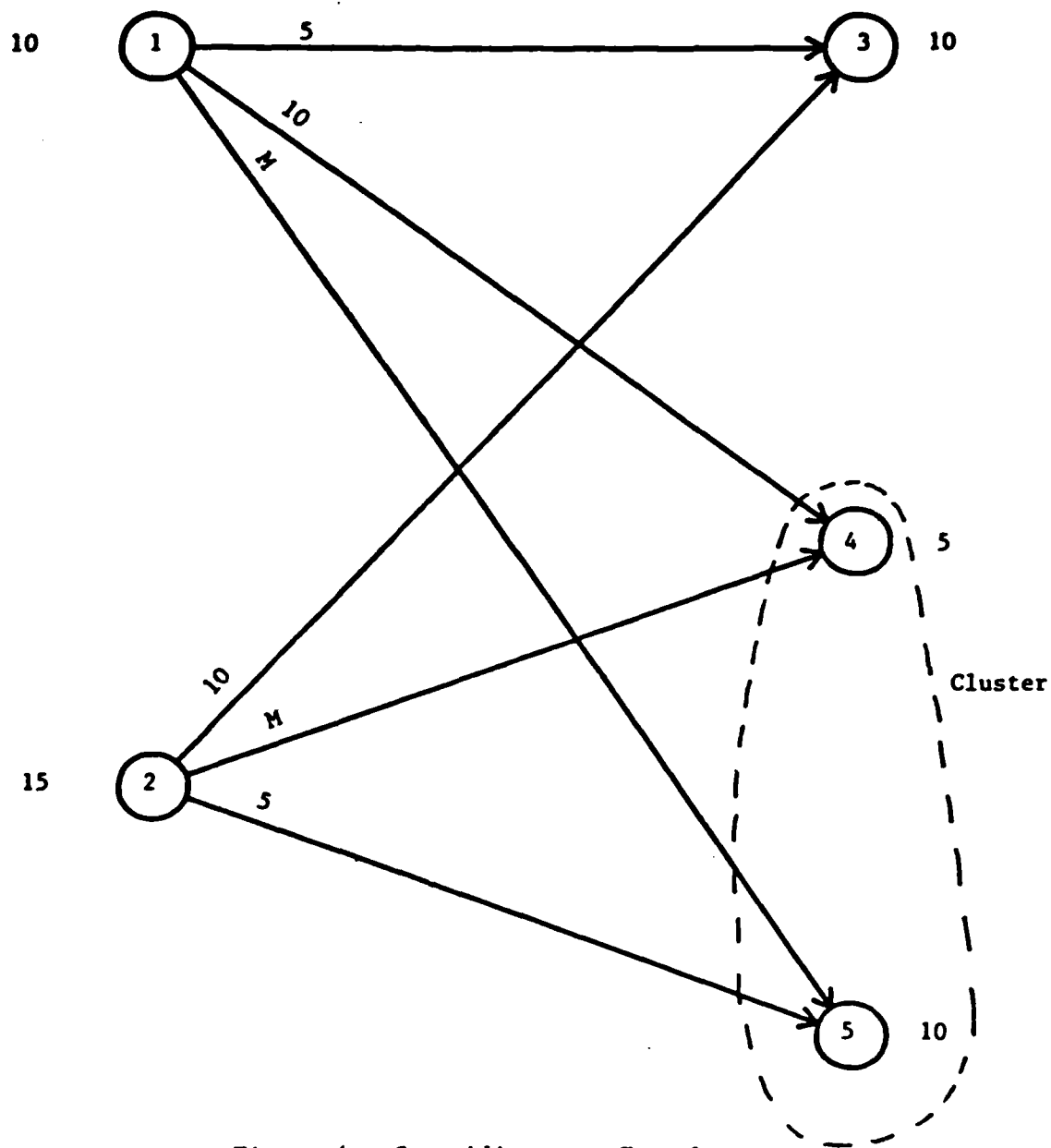


Figure 4. Cost Adjustment Example

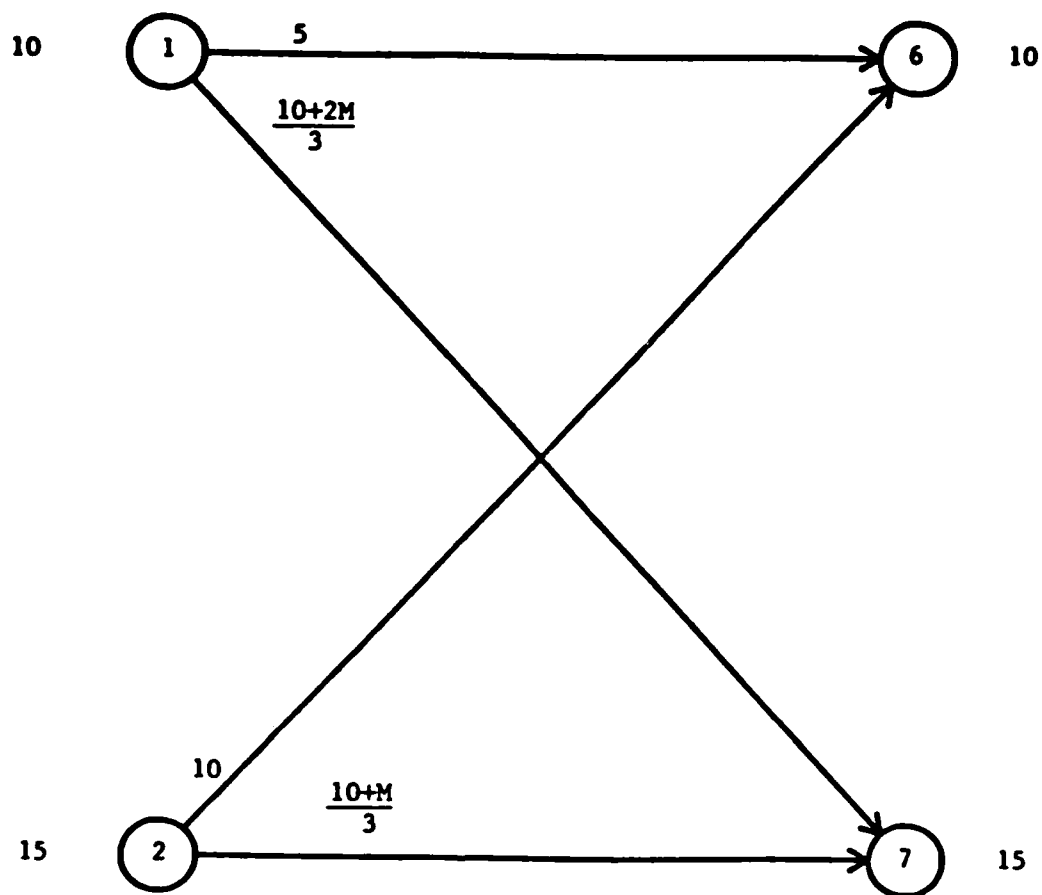


Figure 5. Aggregate Problem with Big-M Costs

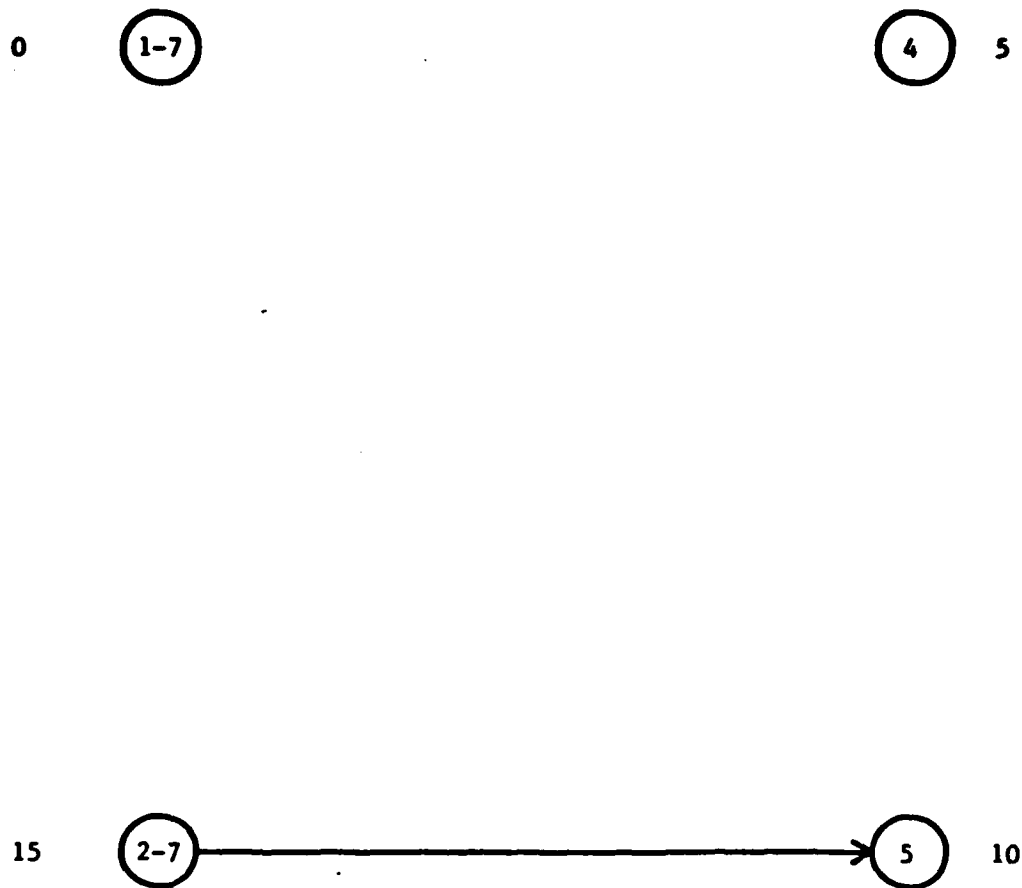


Figure 6. Disaggregation of Cluster Node 7

problem. This problem then becomes a capacitated transportation problem. The capacities may be iteratively modified so as to move towards feasibility of the subproblems and, hence, a feasible solution to MRMATE. A simple approach sets aggregate arc capacity based on the nodes in the cluster to which the source node has arcs. This approach is shown in Figures 7(a) and 7(b). This method can be shown by the example in Fig 8 to still generate infeasible solutions. The aggregate network with arc capacities (as in Figure 7(b)) corresponding to the network in Figure 8 is shown in Figure 9. When two different aggregate arcs have capacity derived from the same node, the aggregate problem loses information regarding which nodes are providing the aggregate capacity. In the example problem in Fig 8, the aggregate arcs from node 1 and node 3 have lost the information that they share their aggregate capacity through sink node 1. A possible solution to the aggregate problem is to send flows of 5 units on arc 1-8 and 15 units on arc 3-8. The disaggregation problem set up in Figure 10 is infeasible due to sparsity of the detailed problem. These arcs cannot all have flow at their upper bounds in the aggregate problem.

3.2.3 Algorithmic Requirement

One of the main requirements is a cohesive format through which all such classes of iterative procedures could be examined. A desirable algorithm to solve the re-formulated problem consisting of the aggregate and disaggregate problems would maintain the network structure of the problems at every iteration while using some mixture of the solutions generated to move

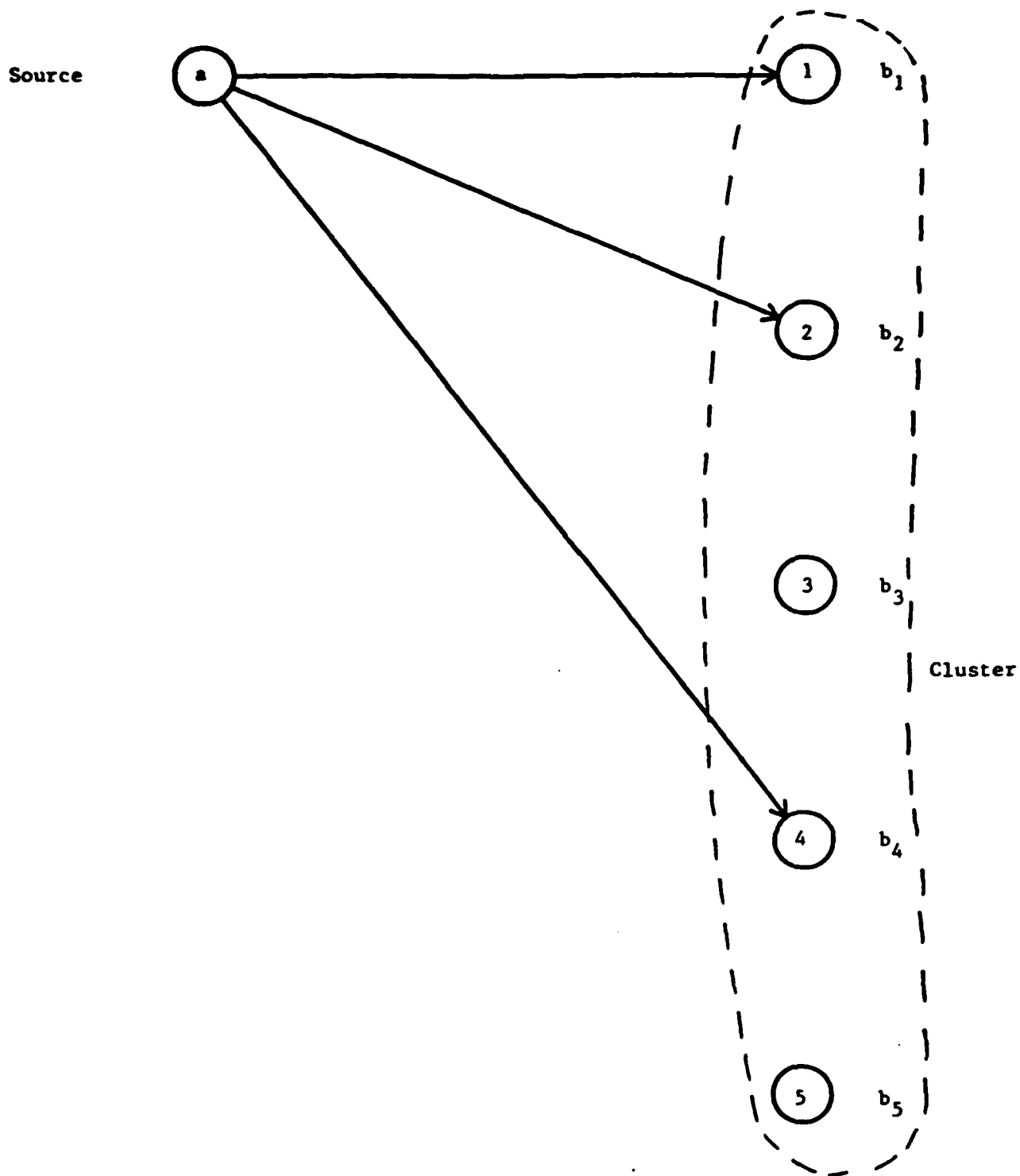


Figure 7(a). Detailed Arcs in a Sparse Network

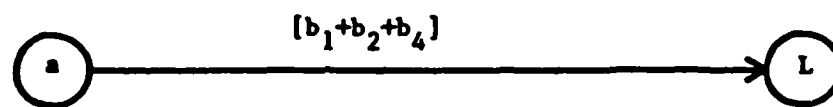


Figure 7(b). Aggregate Arc Capacity

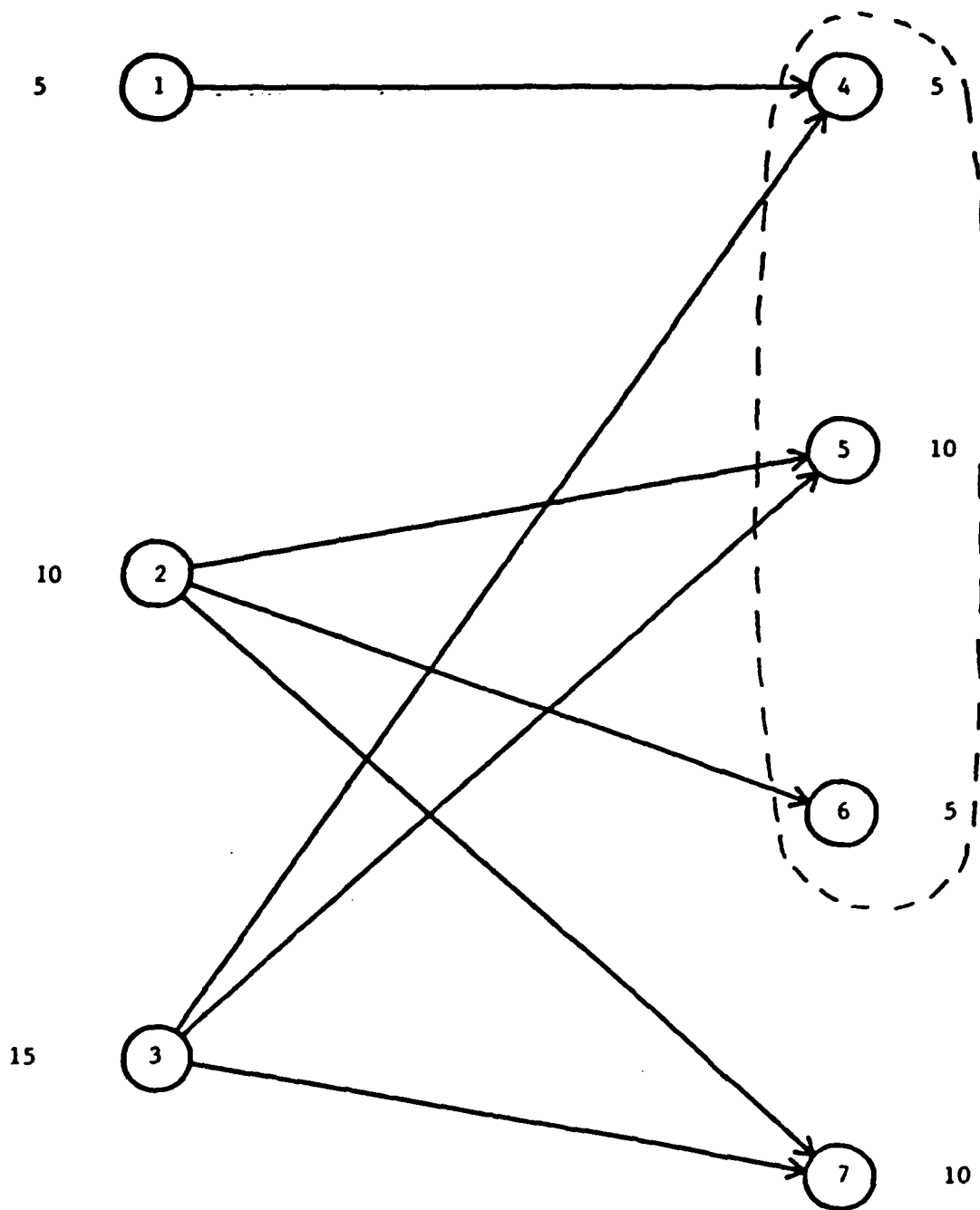


Figure 8. Detailed Network

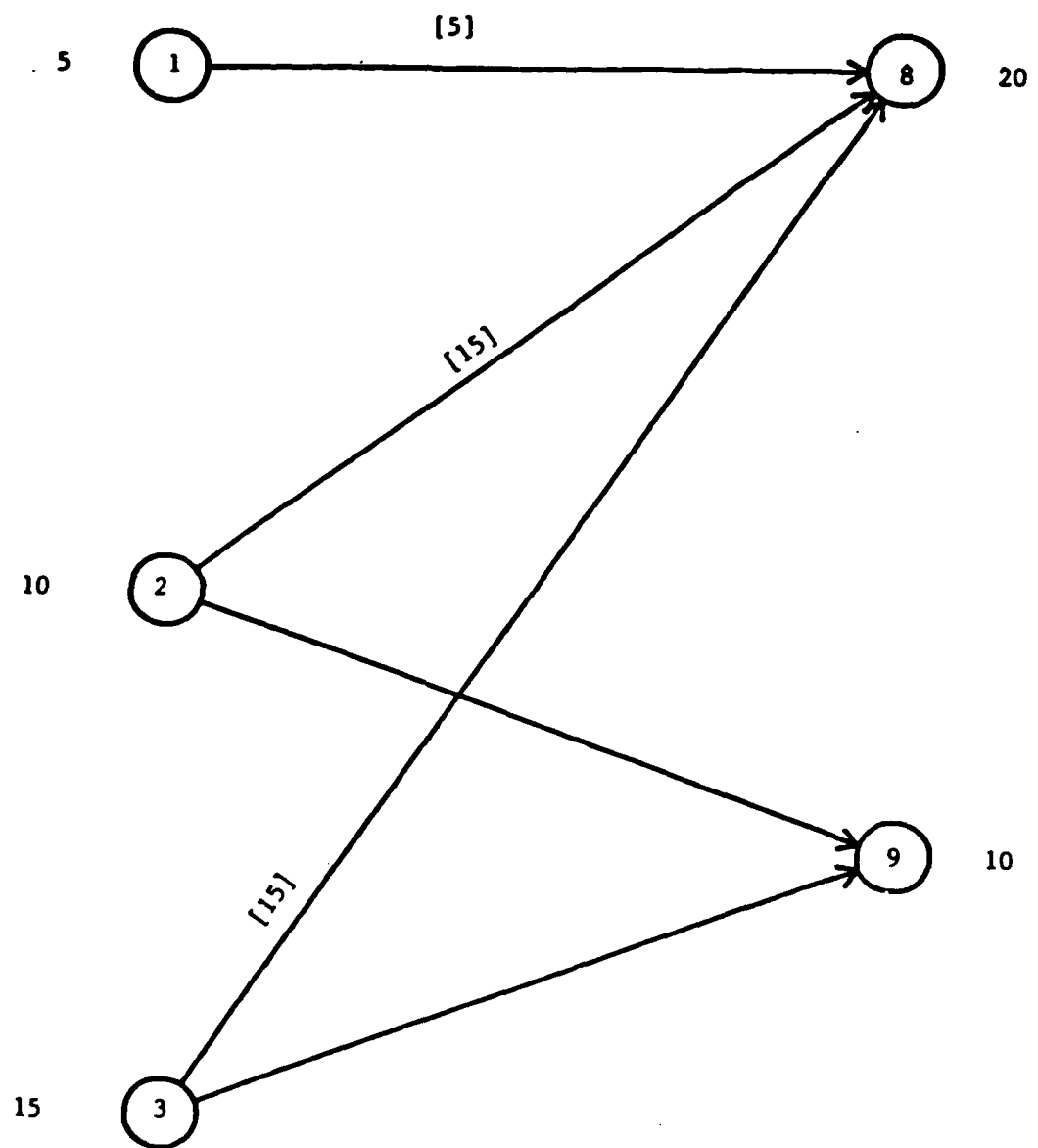


Figure 9. Aggregate Network with Arc Capacities

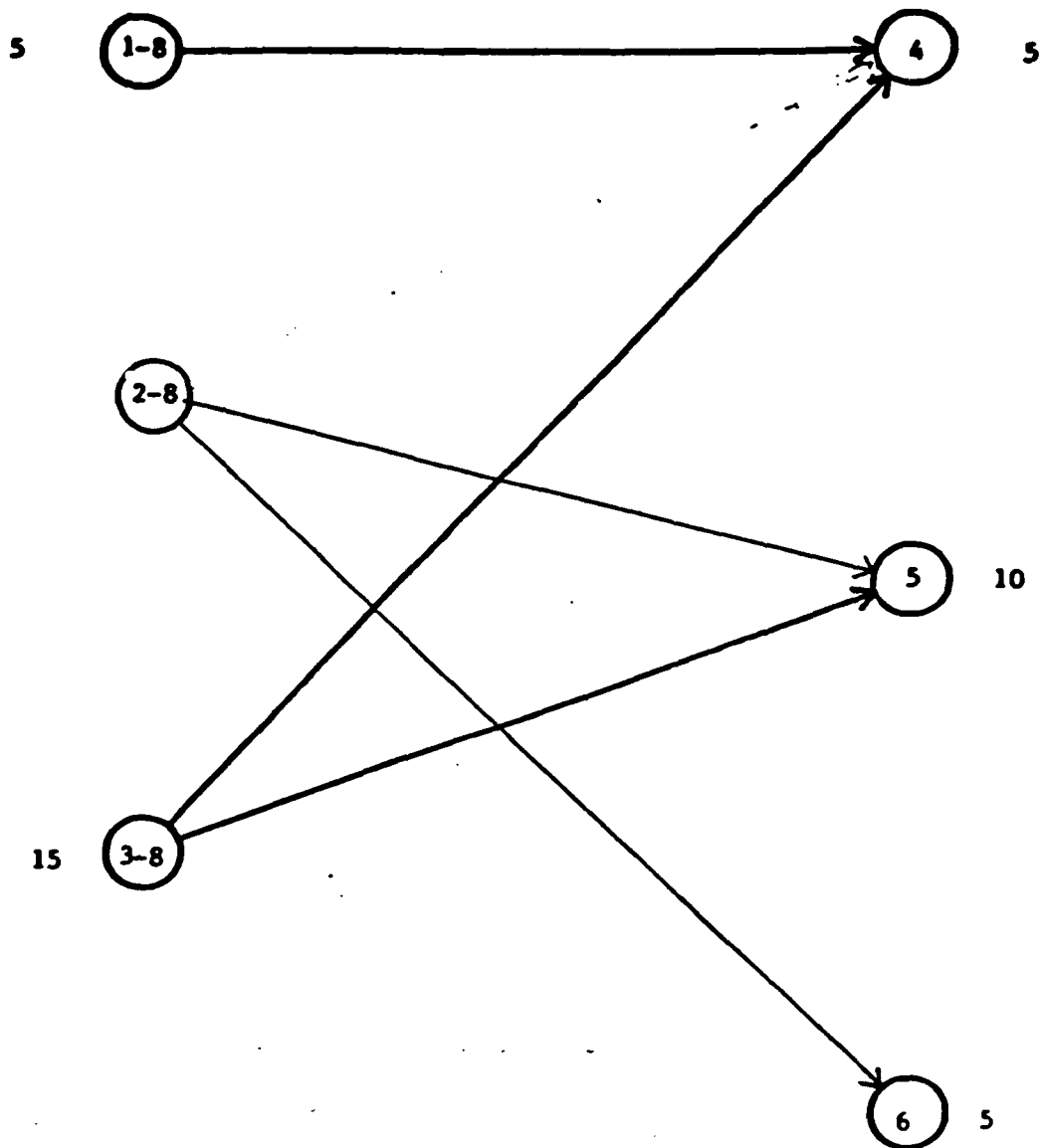


Figure 10. Disaggregation of Cluster Node 8

towards a global optimal solution. A resource allocation procedure in which the aggregate problem sets up resources for the disaggregate problems while the disaggregate problem solutions determine the prices on the aggregate flows would constitute such a desirable procedure.

4.0 TYPES OF MRMATE AGGREGATION

This section discusses different parameters which could be aggregated in MRMATE. Aggregation parameters include aggregation over time, by mode (air/sea), by channels within a time period, and by MRs in a geographical region. An example MRMATE problem is discussed and the effect of different parameter choices on the problem formulation is illustrated.

Consider the MRMATE example in Figure 11. The planning horizon is two periods long. There are four MRs and four channels. The source nodes are the four MRs with the demands equal to the force quantity. The sink nodes are the time expanded channels, hence there are $4 \times 2 = 6$ sink nodes. The channel capability is constant over the planning horizon, so the sink node supplies are determined. MR 1 has to be available on day 1 while MR 2 is available at the channel only on day 2, this information implies that there are no arcs from MR 1 to channels in time period 2 and from MR 2 to channels in time period 1. Also the cargo categories of the MRs permit MR 1 to be shipped on channel 1, MRs 2 and 3 on channels 1 and 2, and MR 4 on any of the channels. The assignments of cargo types to feasible channels at available time periods determines the arcs in the MRMATE network in Figure 11. Also, channels 1 and 3 are given to be air channels, channels 3 and 4 as sea channels. Since the capability of air channels is a weight constraint, that of sea channels is a volume constraint while the MR quantities are expressed in weight units, there is a multiplier associated with each arc to convert the flows to the proper units. The multipliers on arcs to air channels is 1 because the supply and

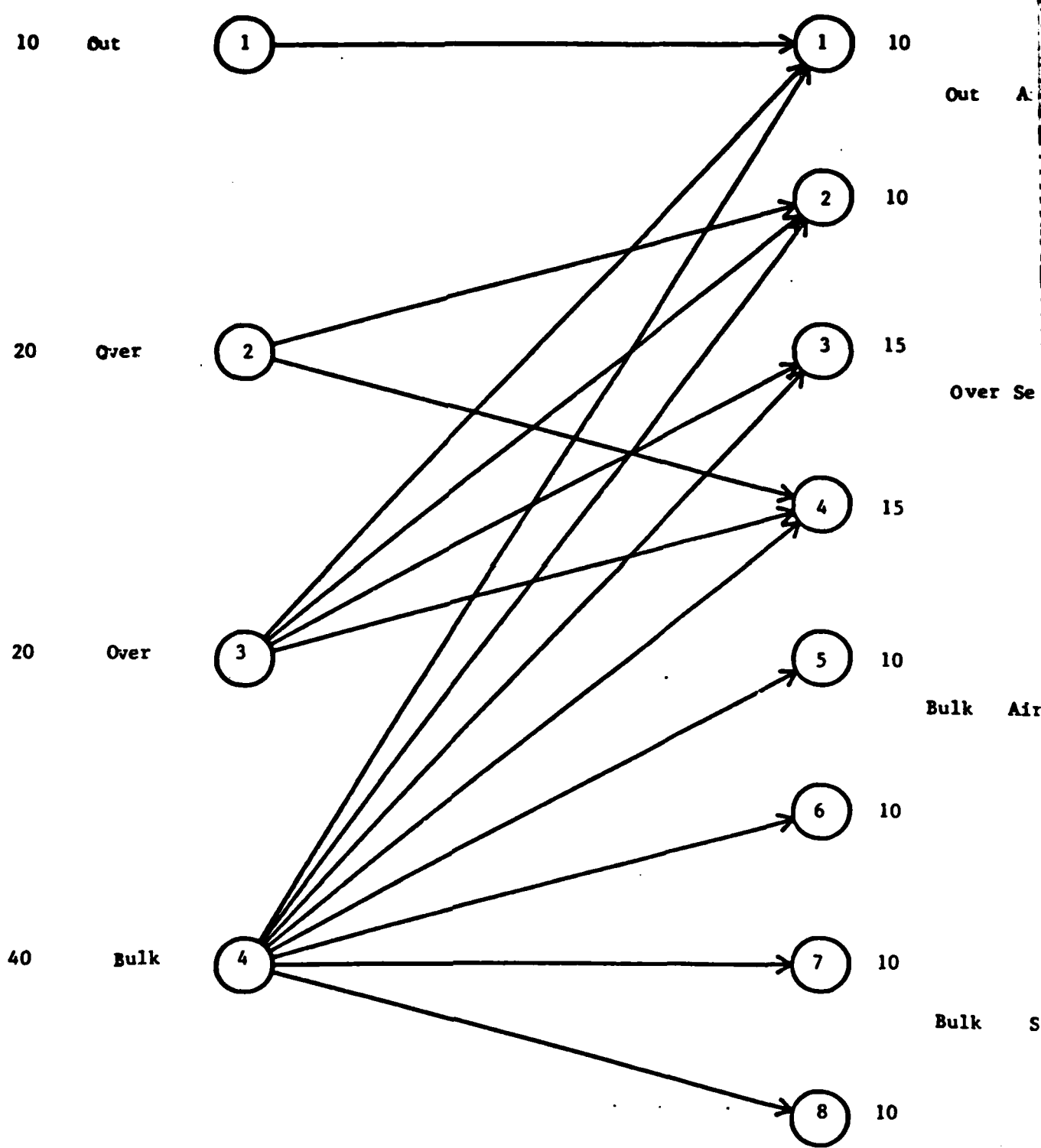


Figure 11. MRMATE Example

demands are in the same units. The multipliers for the four MRs are 1,2,3,2 respectively.

4.1 Aggregation over time

In MRMATE, planning periods and time windows are used to setup channel capabilities over time, and to construct costs of assigning a MR to a channel at a certain time t . Aggregation over time involves creating an aggregate problem by combining information over time. The aggregate problem would have a network structure with the supplies, MRs, connected by arcs to feasible channels based on cargo type. Channel capability is $T * \text{per-unit-time channel capability}$ for each channel. At the aggregate level, time information is absent and the problem becomes one of assigning the MRs to feasible channels. Thus the aggregate problem has

Number of sources = Number of MRs

Number of sinks = Number of channels with non-zero flow as generated by LIFTCAP

Max number of arcs = $M * A * I * J$

where I = number of POEs and J = number of PODs.

The disaggregation problems schedule the MRs, allocated to a channel, across time. The data used to set up these problems are time window information of each MR (providing arc costs) allocation of MRs to the channels at the aggregate level (providing supply information and allocation of channel capability across time). This aggregation procedure can be interpreted as the process of separating the detailed scheduling

information from the assignment decisions of MRs to channels.

There are $(I*J)$ disaggregation problems set up. The aggregate level problem is a generalized network flow problem; the disaggregation problems are pure network flow problems.

It is likely that solving such a series of smaller problems, even though one of them is still a generalized network flow model, will yield smaller overall MRMATE solution times.

4.1.1 Example Problem

For the example problem in Figure 11, aggregation over time yields the aggregate problem as in Figure 12. Arcs are introduced between a MR and an aggregate node if there is at least one detailed arc between that MR and any one of the detailed nodes in the cluster of nodes forming the aggregate. Since MR 1 can be allocated only to channel 1, there are no arcs between MR 1 and any of the other aggregate nodes. The aggregate node capabilities are the sum of the detailed node capabilities as indicated in Figure 12. Since the multiplier associated with all nodes in a cluster are the same, the aggregate arc has the same multiplier as the detailed arcs.

An aggregate problem solution is used to setup the disaggregate problems as in Figures 13(a)-(d) by multiplying the flows by the appropriate multiplier. Thus the aggregate flow between MR 2 and channel 2 is multiplied by 2 to setup the demand in the disaggregate problems in the same units as the detailed node capabilities. Thus detailed channels in disaggregation problems 1 and 3 in Figures 13(a) and 13(c) would have weight units while those in disaggregation problems 2 and 4 in Figures

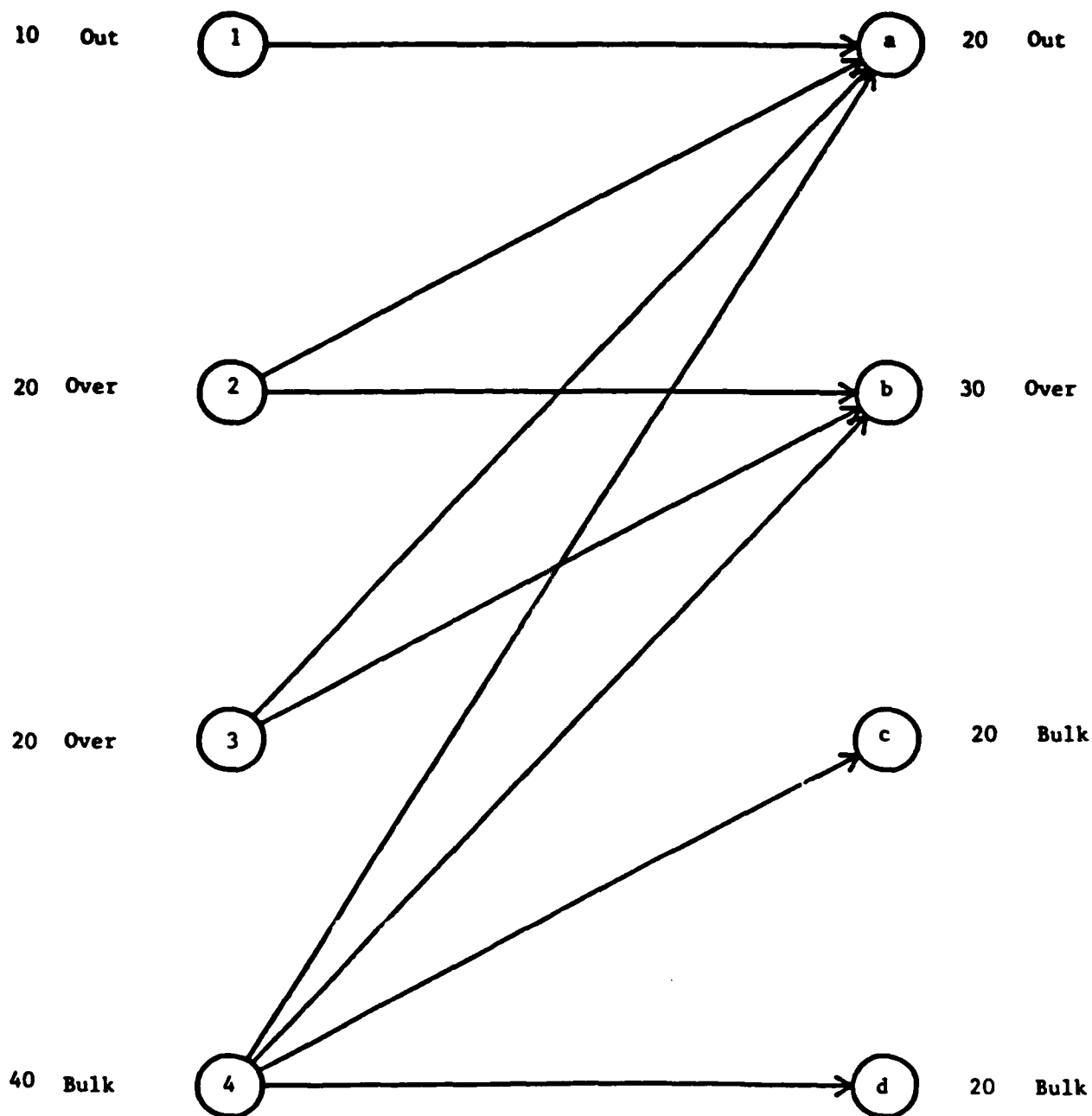


Figure 12. Aggregation over Time

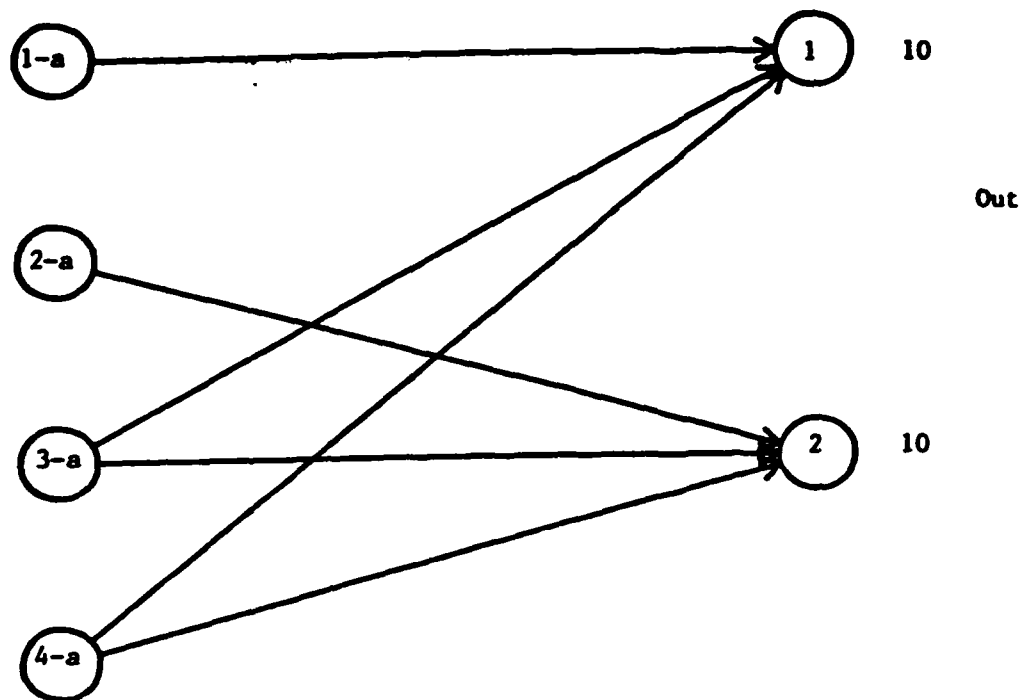


Figure 13(a). Channel 1 Disaggregation

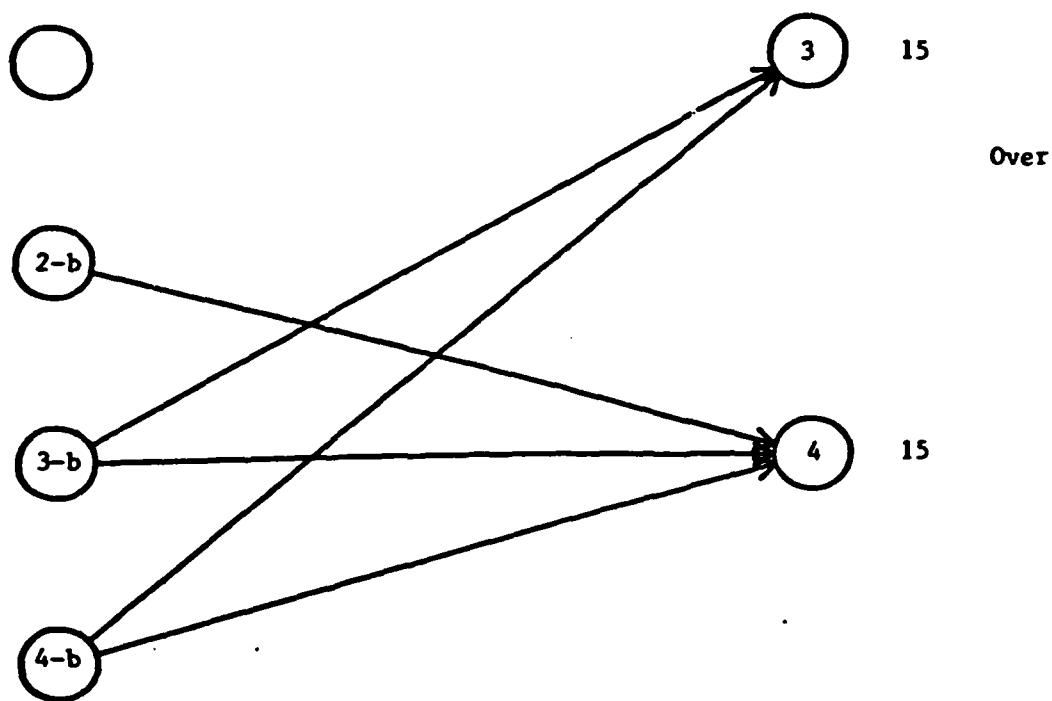


Figure 13(b). Channel 2 Disaggregation

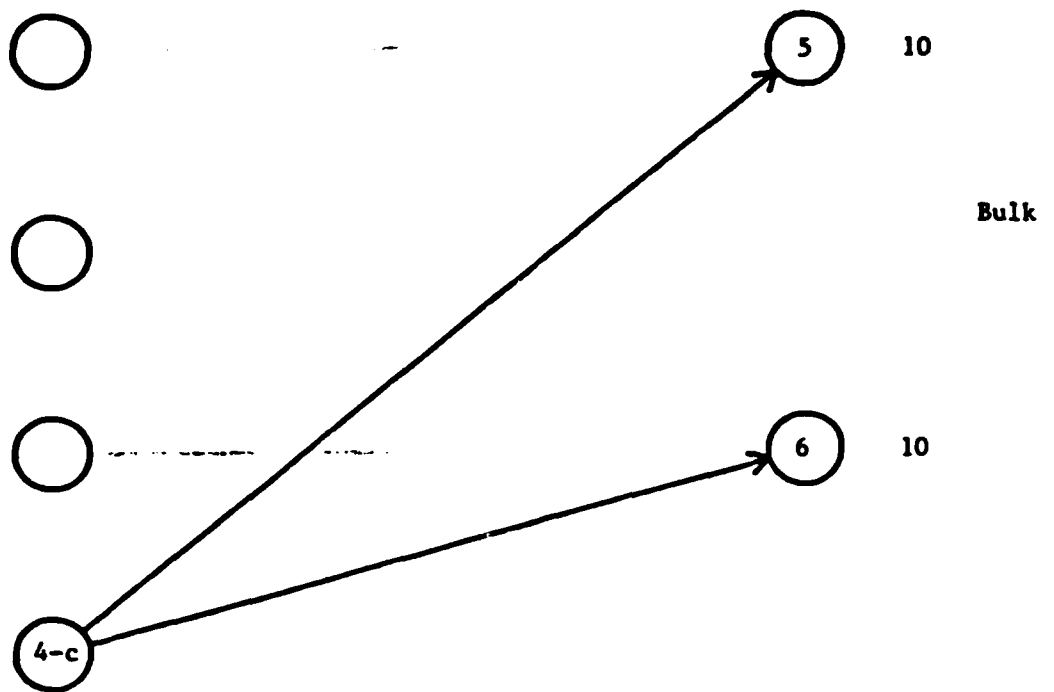


Figure 13(c). Channel 3 Disaggregation

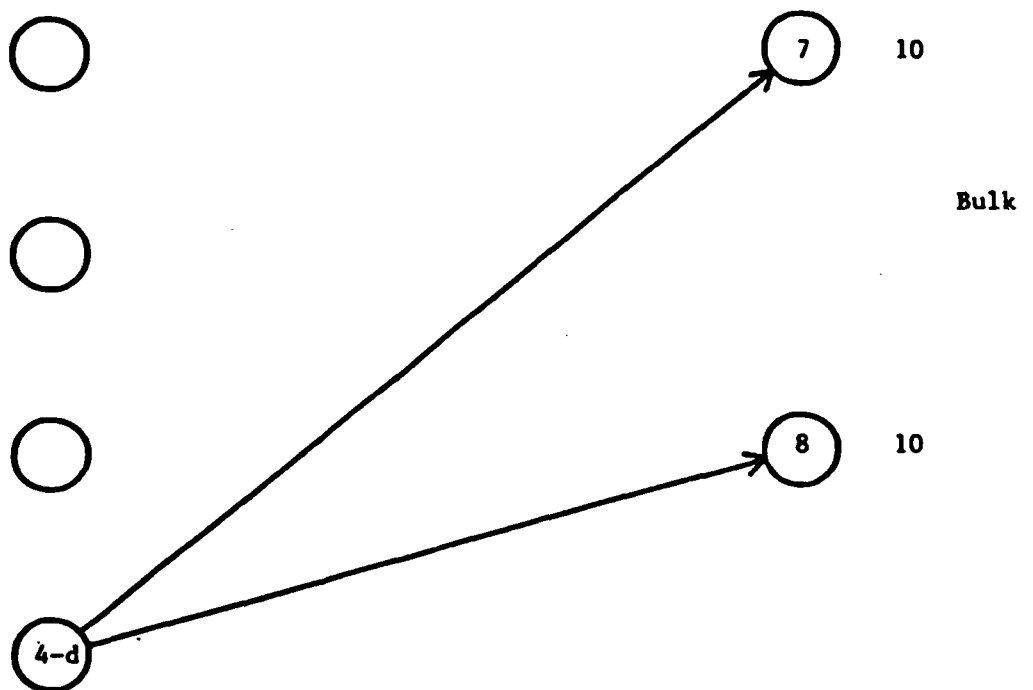


Figure 13(d). Channel 4 Disaggregation

13(b) and 13(d) would be in volume units.

4.2 Aggregation by mode

A second decomposition of MRMATE into a hierarchical structure is provided by aggregating channels which employ a given mode (i.e. air/sea).

The aggregated problem attempts to allocate MRs to each of the two aggregate channels representing air and sea modes. Information on MRs are their supplies. The demand on the two modes is the total capability of all channels across the entire planning period employing the mode. The decision at the aggregate level is the mode split. Information, such as the time windows of MRs or detailed channel capabilities within a mode, is ignored at the aggregate level.

The disaggregation problems make decisions regarding detailed scheduling and allocation within each mode (air/sea). Since the multipliers on arcs for a given mode are the same, the two problems are pure network flow problems. At the disaggregation level, time window information provides costs on the arcs; channel capability and planning period data determine the demand information. Supplies represent the portion of MRs allocated to a mode and are given by the product of aggregate flows and the multiplier for the mode.

This aggregation structure has the added benefit of isolating elements in the problem which affect changes only in a particular mode. Also, since the aggregate problem is a generalized flow problem with only two demand nodes, very fast

procedures could be used to produce a solution. Finally, the disaggregation problems are pure network problems, and this also speeds up the overall solution time.

4.2.1 Example Problem

Aggregation by mode applied to the example in Figure 11 yields the aggregate problem in Figure 14. The aggregate channel capabilities are set up as before. Since all the arcs joining a MR to channels of the same mode have the same multiplier, the aggregate arc multipliers are the same as the multipliers on the detailed arcs. The aggregate problem in Figure 14 is a generalized network flow problem.

Disaggregate problems are set up for each mode as in Figures 15(a) and 15(b). The aggregate flows are multiplied by the appropriate MR multiplier for the sea problem in Figure 15(b), while the air problem uses the aggregate flows in the same units. The disaggregate problems are pure network flow problems.

4.3 Aggregating channels within a time period

Planning periods in MODES refer to periods of time over which channel allocation of assets remain constant. Dividing the planning horizon into T periods has the effect of increasing the number of demand nodes by a factor of T . Also, planning periods infer simultaneous assignment and scheduling decisions in MRMATE. It would be useful to explore aggregation models which separate these two decisions. For example aggregating all channels of same mode in each time period corresponds to making a scheduling decision at an aggregate level and an assignment decision at the

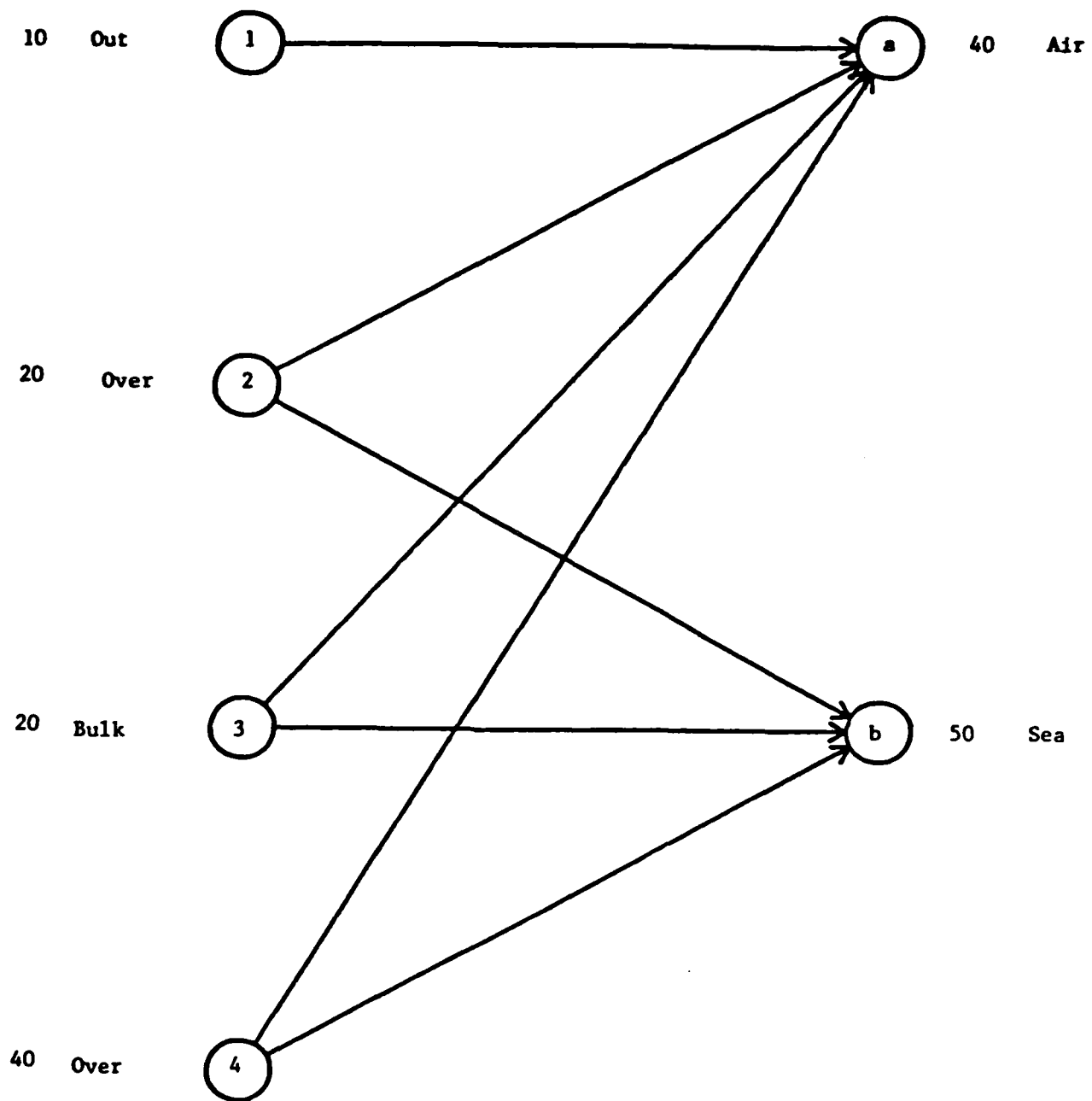


Figure 14. Aggregation by Mode

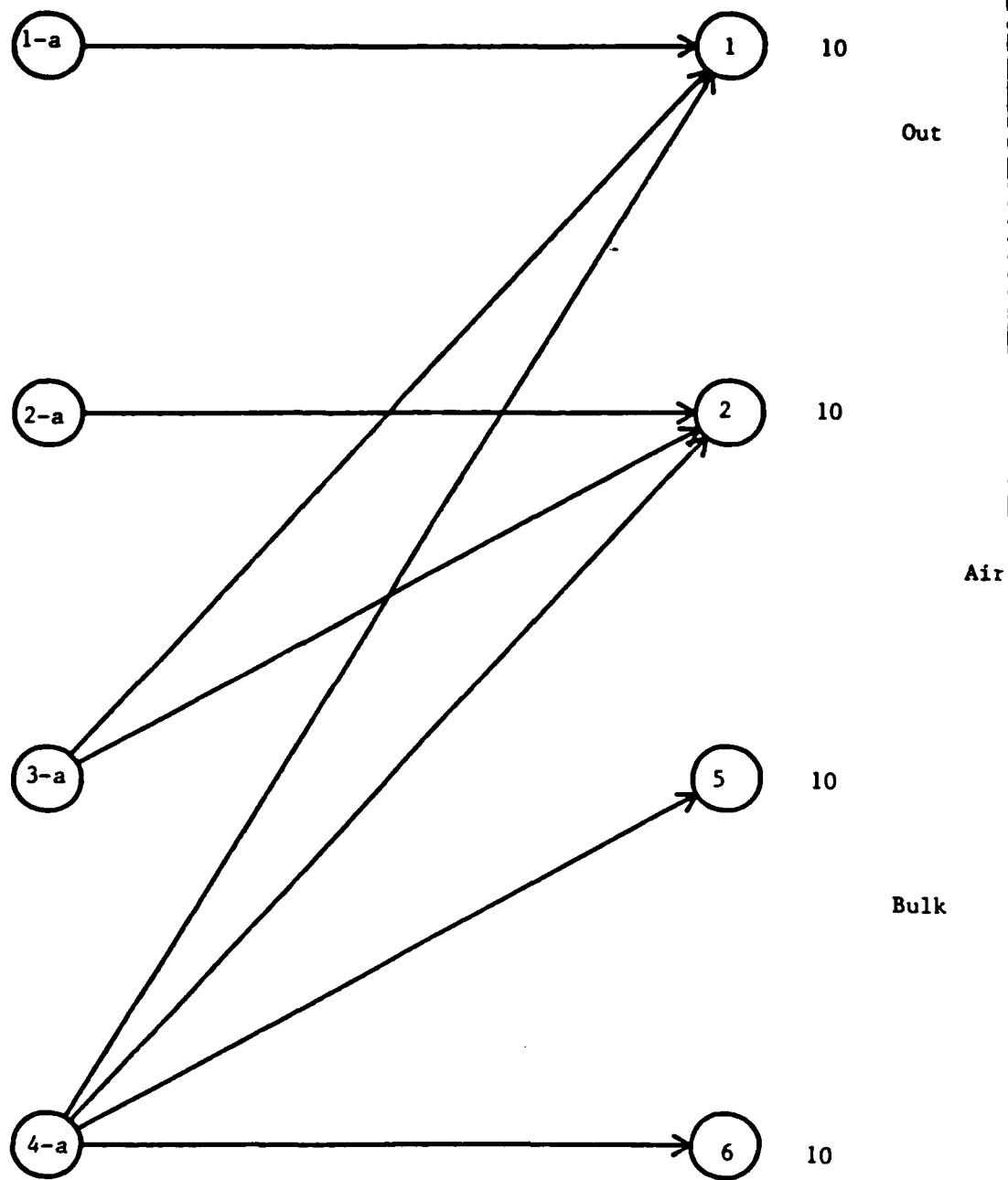


Figure 15(a). Disaggregation Problem for Air Channels

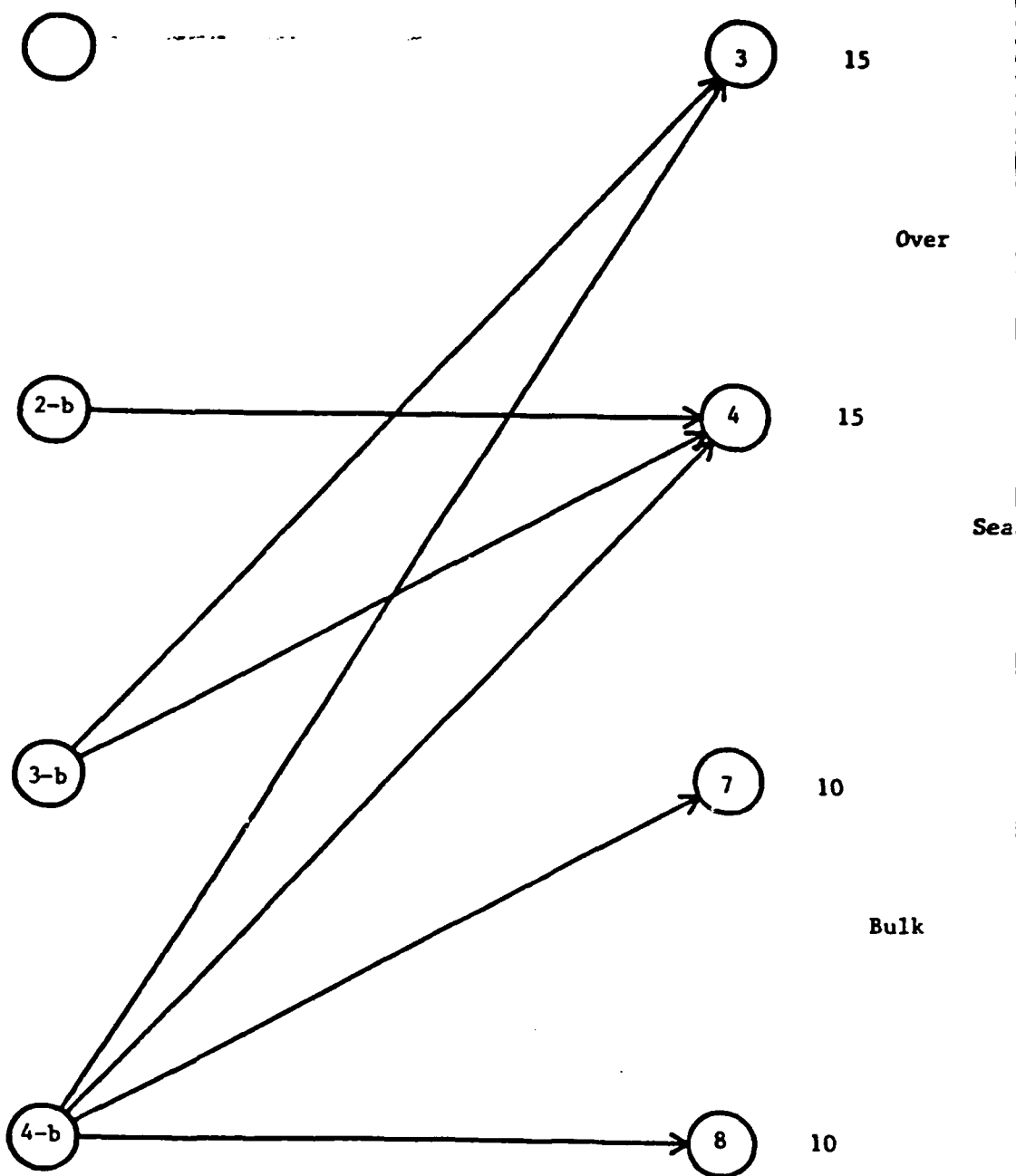


Figure 15(b). Disaggregation Problem for Sea Channels

detailed level. This is the reverse of the decision making roles described in Section 5.1.

The aggregate problem decides which time periods a MR be allocated to. Time window information for each MR is used to setup costs on the aggregate arcs.

At the disaggregate level, $2 * T$ problems are solved to assign MRs shipped during each period to the available channels in a period. This process could be viewed as a process of deciding on a shipping schedule for the MRs for each day of the planning period.

Again the aggregate problem is a generalized network flow problem while the disaggregation problems are pure flow problems. Also, this formulation enables additional constraints, such as unit integrity, to be applied at the second level, i.e. on a day to day basis.

4.3.1 Example Problem

Aggregating channels of the same mode within time periods in the problem in Figure 11 yields an aggregate problem as in Figure 16. Since the channels that are aggregated are of the same mode type, the aggregate arcs have the same multipliers as the detailed arcs. The aggregate problem is a generalized flow problem.

Given the aggregate problem solution $2 * 2 = 4$ disaggregate problems would be set up. The disaggregate problems 1 and 2 in Figures 17(a) and 17(b) would use the aggregate flows in the same units (air multiplier = 1), while in disaggregate problems 3

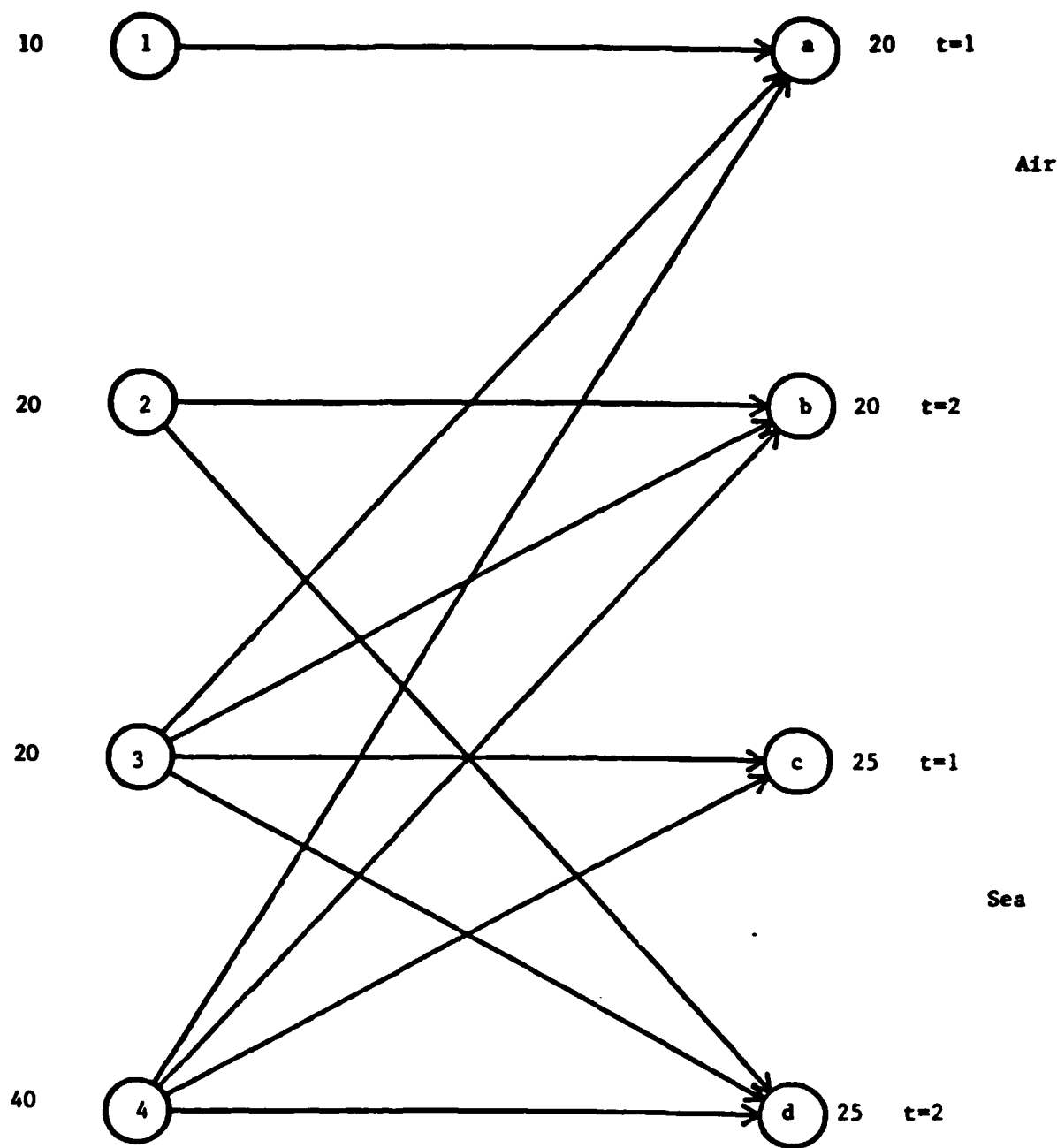


Figure 16. Aggregating Channels by Mode within a Time Period

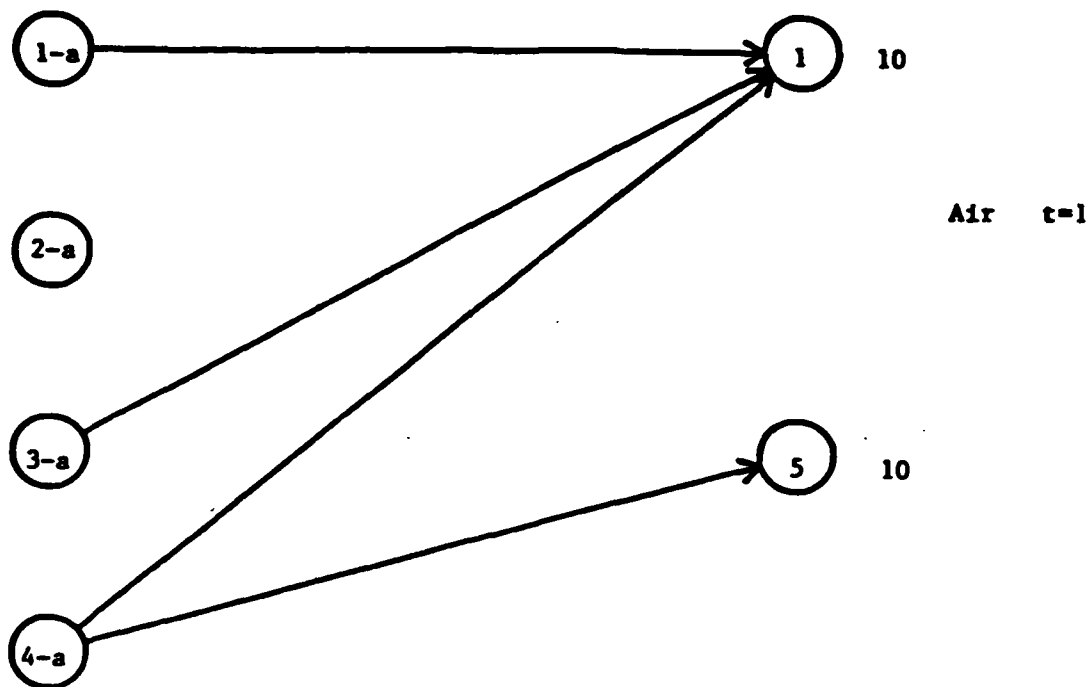


Figure 17(a). Disaggregation Problem for Air at $t=1$

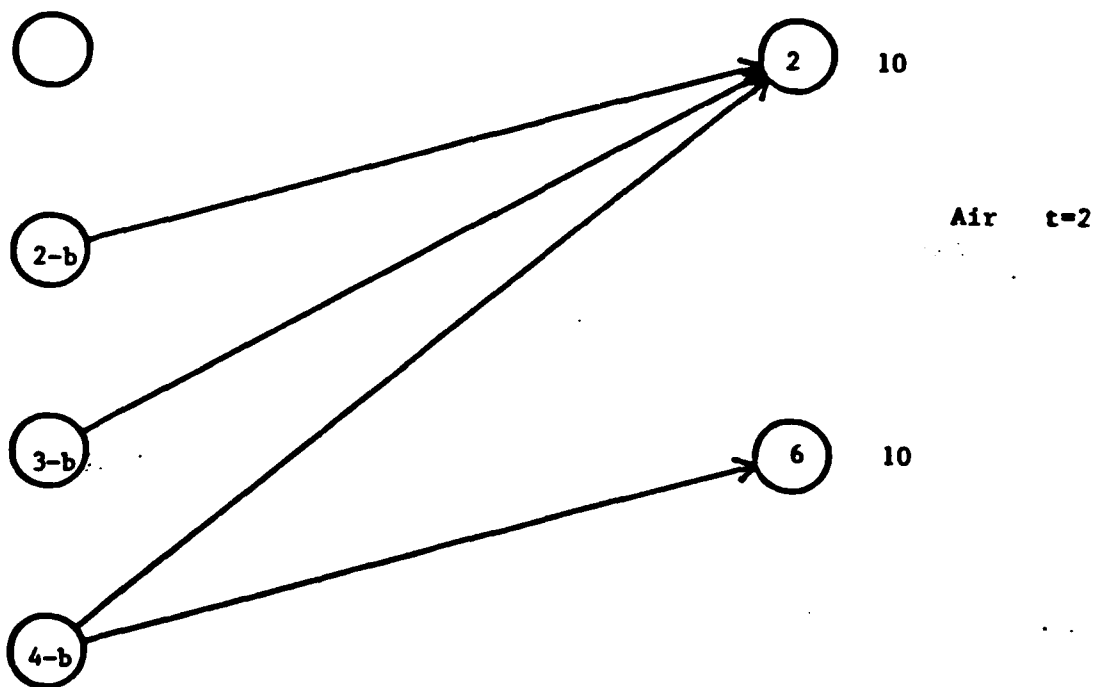


Figure 17(b). Disaggregation Problem for Air at $t=2$

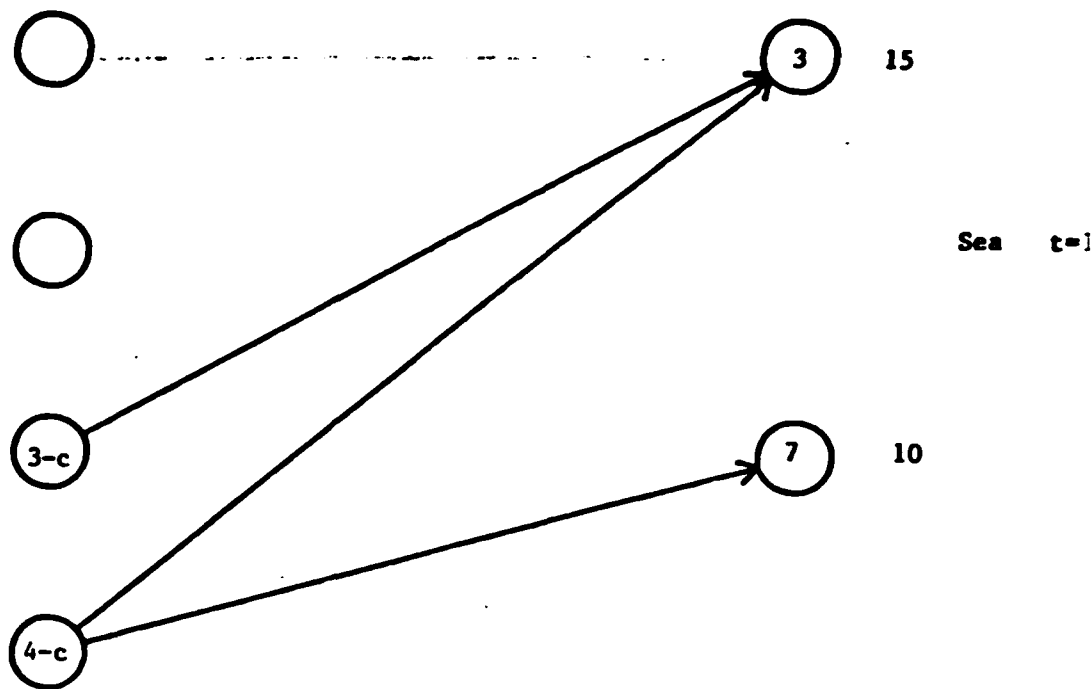


Figure 17(c). Disaggregation Problem for Sea at $t=1$

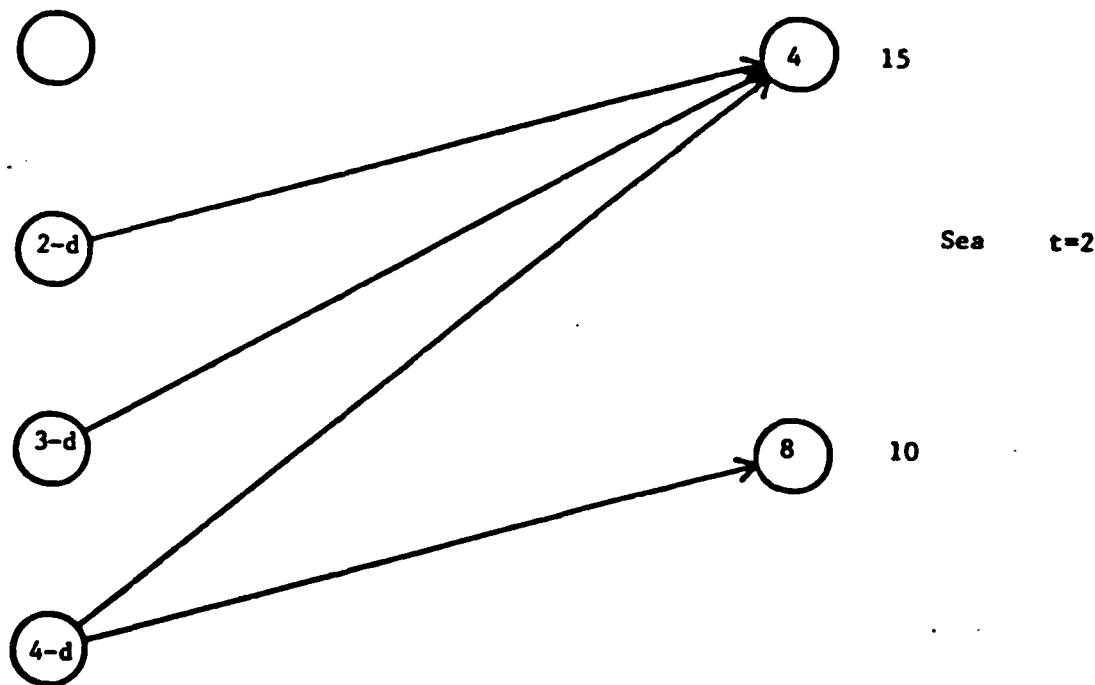


Figure 17(d). Disaggregation Problem for Sea at $t=2$

and 4 in Figures 17(c) and 17(d) the flows are multiplied by the appropriate MR multiplier.

5.0 CONCLUSIONS

This report discussed logical aggregation parameters in MRIMATE. An MRIMATE model formulation was proposed which would utilize these parameters to set up a hierarchical decision making model. Such a model would enable separation of disjoint decision making regions, reduce problem sizes, and facilitate human interface.

Solution techniques for such models are being developed which would maintain the model structure at each level, while moving towards the global optimal solution to the overall MRIMATE problem.

6.0 BIBLIOGRAPHY

1. Balas, E., 'Solution of Large-Scale Transportation Problems through Aggregation', Operations Research, 13, pg 82-93, '65.
2. Geoffrion, A., 'Prior Error Bounds for Procurement Commodity Aggregation in Logistics Planning Models', Naval Research Logistics Quarterly, 24, pg 201-212, '77.
3. Lee, S., 'Surrogate Programming Through Aggregation', Ph.D. Dissertation, UCLA '75.
4. Taylor, R.W., 'Aggregate Programming in Large Scale Linear Systems', Ph.D. Dissertation, Georgia Tech., '83.
5. Zipkin, P.H., 'Aggregation in Linear Programming', Ph.D. Dissertation, Yale University, '77.

END
DTIC

7-86